EXHIBIT Q

Exhibit 9 – Aty et al.

’156 Patent

Claim Limitation (Claim 7)	Exemplary Disclosure
[**156a**] A device comprising:	Aty et al.’s article, *High-Speed, Area-Efficient FPGA-Based Floating-Point Multiplier*, discloses a device. *See, e.g.*: “In this paper, a floating-point multiplier with high speed and area efficient is presented. The multiplier is designed, optimized, and implemented on an FPGA based system.” Aty et al., *High-Speed, Area-Efficient FPGA-Based Floating-Point Multiplier*, at 1.
[**156b**] at least one first low precision high dynamic range (LPHDR) execution unit adapted to execute a first operation on a first input signal representing a first numerical value to produce a first output signal representing a second numerical value,	Aty et al.’s article, *High-Speed, Area-Efficient FPGA-Based Floating-Point Multiplier*, discloses at least one first low precision high dynamic range (LPHDR) execution unit adapted to execute a first operation on a first input signal representing a first numerical value to produce a first output signal representing a second numerical value. *See, e.g.*: “The floating-point multiplier, presented here, is implemented for real-time signal processing on FPGA, which include a 2-D fast Fourier transform (FFT) and a systolic array implementation of an FIR filter. Such signal-processing techniques necessitate a large dynamic range of numbers. The use of floating point helps to alleviate the underflow and overflow problems often seen in fixed-point formats.” Aty et al., *High-Speed, Area-Efficient FPGA-Based Floating-Point Multiplier*, at 1. “Scalable floating-point multiplier allows manipulating range and precision. Hardware description languages (HDL's) [5] are used for this target. Precision and range can be adjusted by controlling the size of mantissa and exponent fields, respectively.” Aty et al., *High-Speed, Area-Efficient FPGA-Based Floating-Point Multiplier*, at 1. “The block diagram of the three-stage, 18-bit floating-point multiplier (m=10, n =7) is shown in Fig. 2. The algorithm for each stage is as follows: Stage 1: • The exponents, e11 and e22 are added and the result along with the carry bit is stored in an 8-bit register. If the addition of two negative exponents results in a value smaller than

Exhibit 9 – Aty et al.

Claim Limitation (Claim 7)	Exemplary Disclosure
	the minimum exponent that can be represented, i.e. -63, underflow occurs. In this case the floating-point number is set to zero. If overflow occurs, the result is set to the maximum number that the format can represent. • If the floating-point number is not zero, the implied one is concatenated to the left side of the f11 and f22 terms • The sign bits are only registered in this stage. • Integer multiplication of the two 11-bit quantities, 1.f22 and 1.f11 is performed. Stage 2: • The output of the integer multiplier will be normalized by shift left until the most significant bit of the mantissa becomes one. • Adjust the exponent depending of the number of the shifting. • The sign bits of the two operands are compared. If they are equal to each other the result is assigned a positive sign, but· if they differ the result is negative. Stage 3: • The output of normalization unit will be rounded to return floating-point value into the correct format. • Remove the bias from the exponent. • The result-sign, exponent and mantissa fields placed into an 18-bit floating point word." Aty et al., *High-Speed, Area-Efficient FPGA-Based Floating-Point Multiplier*, at 2.
[**156c**] wherein the dynamic range of the possible valid inputs to the first operation is at least as wide as from 1/1,000,000 through 1,000,000 and for at least X=5% of the possible valid inputs to the first operation, the statistical mean, over repeated execution of the first operation on each specific input	Aty et al.'s article, *High-Speed, Area-Efficient FPGA-Based Floating-Point Multiplier*, discloses the dynamic range of the possible valid inputs to the first operation is at least as wide as from 1/1,000,000 through 1,000,000 and for at least X=5% of the possible valid inputs to the first operation, the statistical mean, over repeated execution of the first operation on each specific input from the at least X% of the possible valid inputs to the first operation, of the numerical values represented by the first output signal of the LPHDR unit executing the first operation on that input differs by at least Y=0.05% from the result of an exact mathematical calculation of the first operation on the numerical values of that same input. *See, e.g.*:

Exhibit 9 – Aty et al.

Claim Limitation (Claim 7)	Exemplary Disclosure
from the at least X% of the possible valid inputs to the first operation, of the numerical values represented by the first output signal of the LPHDR unit executing the first operation on that input differs by at least Y=0.05% from the result of an exact mathematical calculation of the first operation on the numerical values of that same input; and	“Scalable floating-point multiplier allows manipulating range and precision. Hardware description languages (HDL's) [5] are used for this target. Precision and range can be adjusted by controlling the size of mantissa and exponent fields, respectively.” Aty et al., *High-Speed, Area-Efficient FPGA-Based Floating-Point Multiplier*, at 1. “For comparison purposes, 16 and 18 bits of pipeline floating-point multipliers have been simulated, synthesized, and mapped on Xilinx 4010 using the first configuration which is also used in recently published paper [7]. The code was optimized and the results are given in Table 2. Table 3 shows results from [7]. Comparing the results given in Table 2 and Table 3, it can be seen that the area used by the sixteen bit multiplier in our design is reduced by 7 % and the speed is increased by 81.6%, while for the eighteen bit, the area is reduced by 8% and the speed increased by 155 % with respect to results in [7].” Aty et al., *High-Speed, Area-Efficient FPGA-Based Floating-Point Multiplier*, at 3. (Paper 7 is Shirazi et al.’s *Quantitative Analysis of Floating Point Arithmetic on FPGA Based Custom Computing Machines,*” for which Google has also provided invalidity contentions related to that article.)

Exhibit 9 – Aty et al.

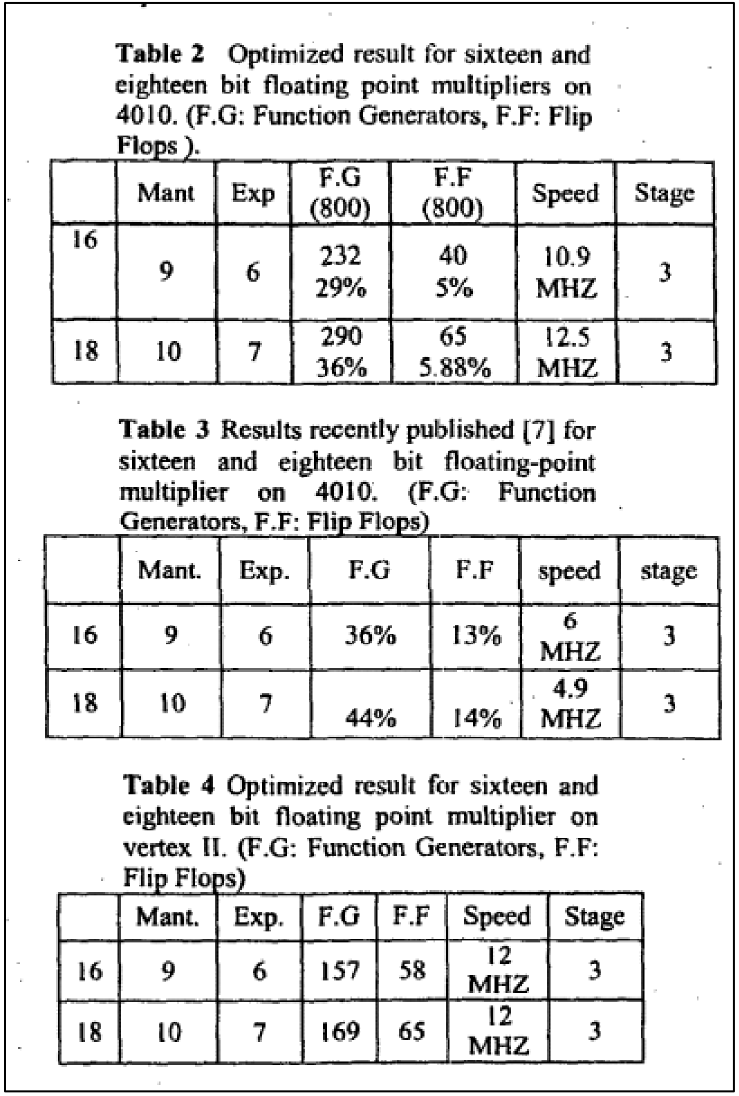
<table>
<tr><th>Claim Limitation (Claim 7)</th><th>Exemplary Disclosure</th></tr>
<tr><td></td><td>


Table 2 Optimized result for sixteen and eighteen bit floating point multipliers on 4010. (F.G: Function Generators, F.F: Flip Flops).

<table>
<tr><th></th><th>Mant</th><th>Exp</th><th>F.G (800)</th><th>F.F (800)</th><th>Speed</th><th>Stage</th></tr>
<tr><td>16</td><td>9</td><td>6</td><td>232
29%</td><td>40
5%</td><td>10.9
MHZ</td><td>3</td></tr>
<tr><td>18</td><td>10</td><td>7</td><td>290
36%</td><td>65
5.88%</td><td>12.5
MHZ</td><td>3</td></tr>
</table>

Table 3 Results recently published [7] for sixteen and eighteen bit floating-point multiplier on 4010. (F.G: Function Generators, F.F: Flip Flops)

<table>
<tr><th></th><th>Mant.</th><th>Exp.</th><th>F.G</th><th>F.F</th><th>speed</th><th>stage</th></tr>
<tr><td>16</td><td>9</td><td>6</td><td>36%</td><td>13%</td><td>6
MHZ</td><td>3</td></tr>
<tr><td>18</td><td>10</td><td>7</td><td>44%</td><td>14%</td><td>4.9
MHZ</td><td>3</td></tr>
</table>

Table 4 Optimized result for sixteen and eighteen bit floating point multiplier on vertex II. (F.G: Function Generators, F.F: Flip Flops)

<table>
<tr><th></th><th>Mant.</th><th>Exp.</th><th>F.G</th><th>F.F</th><th>Speed</th><th>Stage</th></tr>
<tr><td>16</td><td>9</td><td>6</td><td>157</td><td>58</td><td>12
MHZ</td><td>3</td></tr>
<tr><td>18</td><td>10</td><td>7</td><td>169</td><td>65</td><td>12
MHZ</td><td>3</td></tr>
</table>

See also Appendix to Responsive Contentions Regarding Non-Infringement and Invalidity (detailing error rates associated with different mantissa sizes).
</td></tr>
</table>

Exhibit 9 – Aty et al.

Claim Limitation (Claim 7)	Exemplary Disclosure
[**156d**] at least one first computing device adapted to control the operation of the at least one first LPHDR execution unit;	Aty et al.'s article, *High-Speed, Area-Efficient FPGA-Based Floating-Point Multiplier*, discloses at least one first computing device adapted to control the operation of the at least one first LPHDR execution unit. *See, e.g.*: "In this paper, a floating-point multiplier with high speed and area efficient is presented. The multiplier is designed, optimized, and implemented on an FPGA based system." Aty et al., *High-Speed, Area-Efficient FPGA-Based Floating-Point Multiplier*, at 1. "Scalable floating-point multiplier allows manipulating range and precision. Hardware description languages (HDL's) [5] are used for this target. Precision and range can be adjusted by controlling the size of mantissa and exponent fields, respectively." Aty et al., *High-Speed, Area-Efficient FPGA-Based Floating-Point Multiplier*, at 1. To the extent Singular contends that Aty does not disclose a computing device adapted to control the operation of the at least one first LPHDR execution unit, the use of a computing device to control such operations would have been obvious to one skilled in the art. *See* Responsive Contentions Regarding Non-Infringement and Invalidity (describing how every claimed element was obvious to one skilled in the art before Dr. Bates' invention).
[**156e**] wherein the at least one first computing device comprises at least one of a central processing unit (CPU), a graphics processing unit (GPU), a field programmable gate array (FPGA), a microcode-based processor, a hardware sequencer, and a state machine;	Aty et al.'s article, *High-Speed, Area-Efficient FPGA-Based Floating-Point Multiplier*, discloses a device that comprises at least an FPGA. *See, e.g.*: "In this paper, a floating-point multiplier with high speed and area efficient is presented. The multiplier is designed, optimized, and implemented on an FPGA based system." Aty et al., *High-Speed, Area-Efficient FPGA-Based Floating-Point Multiplier*, at 1. Floating-point multipliers with 16, 18, 32,. and 64 bits have been synthesized for Xilinx 4010 FPGAs [4]." Aty et al., *High-Speed, Area-Efficient FPGA-Based Floating-Point Multiplier*, at 1. One of skill in the art would have understood the host computing system would at least include a state machine, as well as a CPU and/or GPU. To the extent Singular contends Aty et al. did

Exhibit 9 – Aty et al.

Claim Limitation (Claim 7)	Exemplary Disclosure
	not disclose at least a microcode-based processor or hardware sequencer, those would have been obvious elements of computing devices adapted to control the operation of the at least one first LPHDR execution units.
[**156f**] and, wherein the number of LPHDR execution units in the device exceeds by at least one hundred the non-negative integer number of execution units in the device adapted to execute at least the operation of multiplication on floating point numbers that are at least 32 bits wide.	Aty et al.'s article, *High-Speed, Area-Efficient FPGA-Based Floating-Point Multiplier*, teaches or suggests a device with multiple matrix multiplication units. *See, e.g.*: "Floating-point multiplication is similar to integer multiplication, because floating-point numbers are stored in sign-magnitude form; the multiplier needs only to deal with unsigned integer numbers and normalization. The floating-point multiplier unit is a three-stage pipeline that produces a result on every clock cycle. The bottleneck of this design was the integer multiplier. There are four different choices for the implementation of the integer multiplier that can be used, namely integer multiplier available in mentor graphic tools with signed or unsigned, array multiplier, parallel multiplier, and series multiplier." Aty et al., *High-Speed, Area-Efficient FPGA-Based Floating-Point Multiplier*, at 2. To the extent Singular contends that Aty et al. did not disclose a device wherein the number of LPHDR execution units exceeds at least one hundred the non-negative integer number of execution units adapted to execute multiplication on floating point numbers that are at least 32 bits wide, that claim would have been obvious to one skilled in the art. *See* Responsive Contentions Regarding Non-Infringement and Invalidity.

Exhibit 9 – Aty et al.

’273 Patent

Claim Limitation (Claim 53)	Exemplary Disclosure
[**273a**] A device:	Aty et al.’s article, *High-Speed, Area-Efficient FPGA-Based Floating-Point Multiplier*, discloses a device. *See* [**156a**].
[**273b**] comprising at least one first low precision high-dynamic range (LPHDR) execution unit adapted to execute a first operation on a first input signal representing a first numerical value to produce a first output signal representing a second numerical value,	Aty et al.’s article, *High-Speed, Area-Efficient FPGA-Based Floating-Point Multiplier*, discloses at least one first low precision high dynamic range (LPHDR) execution unit adapted to execute a first operation on a first input signal representing a first numerical value to produce a first output signal representing a second numerical value. *See* [**156b**].
[**273c**] wherein the dynamic range of the possible valid inputs to the first operation is at least as wide as from 1/1,000,000 through 1,000,000 and for at least X=5% of the possible valid inputs to the first operation, the statistical mean, over repeated execution of the first operation on each specific input from the at least X % of the possible valid inputs to the first operation, of the numerical values represented by the first output signal of the LPHDR unit executing the first operation on that input differs by at least Y=0.05% from the result of an exact mathematical calculation of the first operation on the numerical values of that same input;	Aty et al.’s article, *High-Speed, Area-Efficient FPGA-Based Floating-Point Multiplier*, discloses the dynamic range of the possible valid inputs to the first operation is at least as wide as from 1/1,000,000 through 1,000,000 and for at least X=5% of the possible valid inputs to the first operation, the statistical mean, over repeated execution of the first operation on each specific input from the at least X% of the possible valid inputs to the first operation, of the numerical values represented by the first output signal of the LPHDR unit executing the first operation on that input differs by at least Y=0.05% from the result of an exact mathematical calculation of the first operation on the numerical values of that same input. *See* [**156c**].
[**273d**] wherein the number of LPHDR execution units in the device exceeds by at least one hundred the non-negative integer number of execution units in the device adapted to execute at least the operation of multiplication on floating point numbers that are at least 32 bits wide.	Aty et al.’s article, *High-Speed, Area-Efficient FPGA-Based Floating-Point Multiplier*, teaches or suggests a device with multiple matrix multiplication units. *See* [**156f**].

Exhibit 9 – Aty et al.

’961 Patent

Claim Limitation (Claim 4)	Exemplary Disclosure
[**961a**] A device comprising:	Aty et al.’s article, *High-Speed, Area-Efficient FPGA-Based Floating-Point Multiplier*, discloses a device. *See* [**156a**].
[**961b**] at least one first low precision high-dynamic range (LPHDR) execution unit adapted to execute a first operation on a first input signal representing a first numerical value to produce a first output signal representing a second numerical value,	Aty et al.’s article, *High-Speed, Area-Efficient FPGA-Based Floating-Point Multiplier*, discloses at least one first low precision high dynamic range (LPHDR) execution unit adapted to execute a first operation on a first input signal representing a first numerical value to produce a first output signal representing a second numerical value. *See* [**156b**].
[**961c**] wherein the dynamic range of the possible valid inputs to the first operation is at least as wide as from 1/1,000,000 through 1,000,000 and for at least X=10% of the possible valid inputs to the first operation, the statistical mean, over repeated execution of the first operation on each specific input from the at least X% of the possible valid inputs to the first operation, of the numerical values represented by the first output signal of the LPHDR unit executing the first operation on that input differs by at least Y=0.2% from the result of an exact mathematical calculation of the first operation on the numerical values of that same input; and	Aty et al.’s article, *High-Speed, Area-Efficient FPGA-Based Floating-Point Multiplier*, discloses the dynamic range of the possible valid inputs to the first operation is at least as wide as from 1/1,000,000 through 1,000,000 and for at least X=5% of the possible valid inputs to the first operation, the statistical mean, over repeated execution of the first operation on each specific input from the at least X% of the possible valid inputs to the first operation, of the numerical values represented by the first output signal of the LPHDR unit executing the first operation on that input differs by at least Y=0.05% from the result of an exact mathematical calculation of the first operation on the numerical values of that same input. *See* [**156c].**
[**961d**] at least one first computing device adapted to control the operation of the at least one first LPHDR execution unit.	Aty et al.’s article, *High-Speed, Area-Efficient FPGA-Based Floating-Point Multiplier*, discloses at least one first computing device adapted to control the operation of the at least one first LPHDR execution unit. *See* [**156d**].

Exhibit 9 – Aty et al.

Claim Limitation (Claim 13)	Exemplary Disclosure
[**961e**] A device comprising:	Aty et al.'s article, *High-Speed, Area-Efficient FPGA-Based Floating-Point Multiplier*, discloses a device. *See* [**156a**].
[**961f**] a plurality of components comprising:	Aty et al.'s article, *High-Speed, Area-Efficient FPGA-Based Floating-Point Multiplier*, discloses at least one first low precision high dynamic range (LPHDR) execution unit adapted to execute a first operation on a first input signal representing a first numerical value to produce a first output signal representing a second numerical value. *See* [**156b].** Aty et al.'s article, *High-Speed, Area-Efficient FPGA-Based Floating-Point Multiplier*, discloses at least one first computing device adapted to control the operation of the at least one first LPHDR execution unit. *See* [**156d**].
[**961g**] at least one first low precision high-dynamic range (LPHDR) execution unit adapted to execute a first operation on a first input signal representing a first numerical value to produce a first output signal representing a second numerical value,	Aty et al.'s article, *High-Speed, Area-Efficient FPGA-Based Floating-Point Multiplier*, discloses at least one first low precision high dynamic range (LPHDR) execution unit adapted to execute a first operation on a first input signal representing a first numerical value to produce a first output signal representing a second numerical value. *See* [**156b**].
[**961h**] wherein the dynamic range of the possible valid inputs to the first operation is at least as wide as from 1/1,000,000 through 1,000,000 and for at least X=10% of the possible valid inputs to the first operation, the statistical mean, over repeated execution of the first operation on each specific input from the at least X% of the possible valid inputs to the first operation, of the numerical values represented by the first output signal of the LPHDR unit executing the first operation on that input differs by at least Y=0.2% from the result of an exact mathematical calculation of the first operation on the numerical values of that same input.	Aty et al.'s article, *High-Speed, Area-Efficient FPGA-Based Floating-Point Multiplier*, discloses the dynamic range of the possible valid inputs to the first operation is at least as wide as from 1/1,000,000 through 1,000,000 and for at least X=5% of the possible valid inputs to the first operation, the statistical mean, over repeated execution of the first operation on each specific input from the at least X% of the possible valid inputs to the first operation, of the numerical values represented by the first output signal of the LPHDR unit executing the first operation on that input differs by at least Y=0.05% from the result of an exact mathematical calculation of the first operation on the numerical values of that same input. *See* [**156c**].